

3-HITTING SET on bounded degree hypergraphs: Upper and lower bounds on the kernel size

Iyad A. Kanj^{1,*}

Fenghui Zhang²

¹ School of Computing, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604, USA. ikanj@cs.depaul.edu

² Google Kirkland, 747 6th Street South, Kirkland, WA 98033, USA. fhzhang@gmail.com

Abstract. We study upper and lower bounds on the kernel size for the 3-HITTING SET problem on hypergraphs of degree at most 3, denoted 3-3-HS. We first show that, unless $P=NP$, 3-3-HS on 3-uniform hypergraphs does not have a kernel of size at most $35k/19 > 1.8421k$. We then give a $4k - k^{0.2692}$ kernel for 3-3-HS that is computable in time $O(k^{1.2692})$.¹ This result improves the upper bound of $4k$ on the kernel size for 3-3-HS, given by Wahlström. We also show that the upper bound results on the kernel size for 3-3-HS can be generalized to the 3-HS problem on hypergraphs of bounded degree Δ , for any integer-constant $\Delta > 3$.

Keywords. hitting set, kernel, upper bounds, lower bounds, parameterized complexity

1 Introduction

A *hitting set* in a hypergraph $\mathcal{H} = (V, E)$ is a set of vertices S such that every hyperedge in \mathcal{H} contains at least one vertex from S . The *size* of a hitting set S is $|S|$. The HITTING SET problem is: Given a hypergraph $\mathcal{H} = (V, E)$ and a nonnegative integer k , decide if there exists a hitting set for \mathcal{H} of size at most k . Therefore, the HITTING SET problem is a generalization of the VERTEX COVER problem (given a graph G and a nonnegative integer k , decide if there exists a subset of vertices C in G such that every edge in G is incident on at least one vertex of C) to hypergraphs. The VERTEX COVER problem, one of the first few problems proven to be NP-complete [12], has very important applications in different areas of science and engineering, where it usually models conflict-resolution problems, and has received considerable interest from researchers in several areas of theoretical computer science (approximation, exact/parameterized algorithms, kernelization, etc).

* The work of this author was partially supported by a DePaul University Competitive Research Grant.

¹ We do not assume that the hypergraph is 3-uniform for the kernel upper bound results.

The HITTING SET problem seems to be much more difficult than the VERTEX COVER problem, at least from the approximation theory and parameterized complexity points of view. Whereas the VERTEX COVER problem can be approximated (in polynomial time) to within ratio 2, the HITTING SET problem is not approximable to within ratio $c \lg n$ for some constant $c > 0$, where n is the number of vertices in the hypergraph [2]. From the parameterized complexity perspective, whereas the VERTEX COVER problem is fixed-parameter tractable, the HITTING SET problem is W[2]-complete [10], and hence is unlikely to be solvable in time $f(k)n^{O(1)}$, for any recursive function f .

The parameterized intractability of the HITTING SET problem led researchers in parameterized complexity to consider the special case in which the size of the hyperedge (the number of vertices in the hyperedge) is at most d , for some integer constant d ; this problem is referred to as the d -HITTING SET problem, and can be easily seen to be fixed-parameter tractable. In particular, the 3-HITTING SET problem, denoted 3-HS, received a lot of attention. Niedermeier and Rosmanith [13] gave an algorithm for 3-HS running in time $O^*(2.27^k)$.² Subsequently, Fernau [11] improved Niedermeier and Rosmanith’s result by giving an algorithm for 3-HS that runs in time $O^*(2.179^k)$. This result was further improved by Wahlström [15], who gave an algorithm for 3-HS that runs in time $O^*(2.07^k)$ [15]. In terms of kernelization, the 3-HS problem was shown to admit a kernel of size $O(k^3)$ [13]. This result was recently improved by Abu-Khzam, who gave a kernel of size $O(k^2)$ for 3-HS [1]; this upper bound currently stands as the best upper bound on the kernel size for 3-HS. Therefore, in contrast to the VERTEX COVER problem, which is equivalent to the 2-HS problem and hence is a special case of 3-HS, the 3-HS problem seems to be more difficult—modulo parameterized complexity—than VERTEX COVER: Whereas VERTEX COVER is solvable in time $O^*(1.274^k)$ [9] and admits a kernel of size $2k$ [3, 7], the currently-best algorithm for 3-HS runs in time $O^*(2.07^k)$ [15], and the currently-best upper bound on the kernel size for 3-HS is $O(k^2)$ [1].

The 3-HS problem on bounded-degree hypergraphs has been considered as well, especially from the kernelization point of view.³ Wahlström [15] showed that the 3-HS problem on hypergraphs of degree at most Δ has a kernel of size at most $(\Delta + 1)k$. He also claimed a lower bound of $(\Delta + 5)k/(\Delta + 1)$ on the size of the kernel for the problem [15]. However, the proof that he provided is flawed. Recently, under the assumption that $P \neq NP$, Cai proved a lower bound of $(1 + 1/(\sqrt{2\Delta + 1} - 1) - \epsilon)k$, for any $\epsilon > 0$, on the size of the kernel for the 3-HS problem on 3-uniform hypergraphs of degree at most Δ [4]. Cai also gave upper and lower bounds on the kernel size for the 3-HS problem on planar hypergraphs [4].

In this paper we study upper and lower bounds on the kernel size for 3-HS on hypergraphs of degree at most 3, denoted 3-3-HS. We note that the VERTEX

² The asymptotic notation $O^*(f(k))$ denotes time complexity of the form $f(k) \cdot p()$, where p is a polynomial of the input size.

³ Clearly the problem remains NP-complete because VERTEX COVER on graphs of maximum degree at most 3 is NP-complete [12].

COVER problem on graphs of degree at most 3, denoted VC-3, which is a special case of 3-3-HS, has received considerable attention from the parameterized complexity point of view (for example, see [7, 8, 14]). We start by showing that, unless $P=NP$, the 3-3-HS problem on 3-uniform hypergraphs does not have a kernel of size at most $35k/19 > 1.8421k$; this improves the lower bound result of $(1 + 1/(\sqrt{7} - 1) - \epsilon)k < 1.608k$, given by Cai [4].

We then present an algorithm that computes a kernel of size at most $4k - k^{0.2692}$ for 3-3-HS, and that runs in time $O(k^{1.2692})$. (We note that for these results we do not assume that the hypergraph is 3-uniform.) This improves the upper bound of $4k$ on the kernel size for 3-3-HS, given by Wahlström [15]. We note that, even though the improvement in the upper bound for the kernel size is small, the techniques involved rely on deep structural observations. We also show that these upper bound results can be generalized to the 3-HS problem on hypergraphs of degree at most Δ , for any integer constant $\Delta > 3$.

2 Preliminaries

We describe some of the notations and terminologies used in the paper. The reader is referred to Downey and Fellows' book [10] for more details about parameterized complexity theory, and to West [16] for more information on graphs and hypergraphs.

For a graph G , $V(G)$ and $E(G)$ are the sets of vertices and edges of G ; $n(G) = |V(G)|$ and $e(G) = |E(G)|$ are the number of vertices and edges in G . For a vertex v , we let $N(v)$ be the set of vertices adjacent to v . The *degree* of a vertex v , $\deg(v)$, is the number of edges incident to v in G . The subgraph $G - v$ of G is obtained from G by removing $v \in V(G)$ and its incident edges.

A *hypergraph* $\mathcal{H} = (V, E)$ consists of a *vertex set* $V = V(\mathcal{H})$ and an *edge set* $E = E(\mathcal{H})$ so that $e \subseteq V$ for every $e \in E$. The *degree* of a vertex v in \mathcal{H} , denoted $\deg(v)$, is defined as the number of edges in E that contain v . Two vertices u and v are *adjacent* or *neighbors* in \mathcal{H} if there exists an edge $e \in \mathcal{H}$ such that $\{u, v\} \subseteq e$. The *distance* between two vertices u and v in \mathcal{H} is the length of a shortest path between u and v in \mathcal{H} , where a path in \mathcal{H} is naturally defined using the adjacency relationship described above. An edge e is called an *i -edge* if $|e| = i$. A hypergraph \mathcal{H} is *i -uniform* if every edge in \mathcal{H} is an i -edge. We say that a vertex u *dominates* a vertex v if every edge containing v also contains u . A set of vertices Γ is said to be *dominated* by a set of vertices Γ' if every edge in \mathcal{H} containing a vertex of Γ also contains a vertex of Γ' . An edge e is said to *dominate* another edge e' in \mathcal{H} if e' is a subset of e (i.e., $e' \subseteq e$). For a vertex $v \in \mathcal{H}$, $\mathcal{H} - v$ is the hypergraph resulting from removing vertex v from every edge in \mathcal{H} . For an edge $e \in \mathcal{H}$, $\mathcal{H} - e$ is the hypergraph resulting from removing edge e from \mathcal{H} (note that the vertices contained in e remain in \mathcal{H}).

A *hitting set* in a hypergraph $\mathcal{H} = (V, E)$ is a set of vertices S such that every hyperedge in \mathcal{H} contains at least one vertex from S . The *size* of a hitting set S is $|S|$. A hitting set S is *minimum* if its size is minimum among all hitting sets of \mathcal{H} . The HITTING SET problem is: Given a hypergraph $\mathcal{H} = (V, E)$ and a

nonnegative integer k , decide if there exists a hitting set S of \mathcal{H} whose size is at most k . The 3-HITTING SET problem, denoted 3-HS, is the set of instances (\mathcal{H}, k) of the HITTING SET problem in which every edge in \mathcal{H} has cardinality at most 3. For an integer constant $\Delta \geq 2$, the Δ -3-HS problem refers to the 3-HS problem on hypergraphs of degree at most Δ .

A *parameterized problem* is a set of instances of the form (x, k) , where $x \in \Sigma^*$ for a finite alphabet set Σ , and k is a non-negative integer called the *parameter*. A parameterized problem Q is *fixed parameter tractable*, or simply FPT, if there exists an algorithm that on input (x, k) decides if (x, k) is a yes-instance of Q in time $f(k)n^{O(1)}$, where f is a recursive function independent of $n = |x|$. A parameterized problem Q is *kernelizable* if there exists a polynomial-time reduction that maps an instance (x, k) of Q to another instance (x', k') of Q such that: (1) $|x'| \leq g(k)$ for some recursive function g , (2) $k' \leq f(k)$ for some recursive function f , and (3) (x, k) is a yes-instance of Q if and only if (x', k') is a yes-instance of Q . The instance x' is called the *kernel* of x .

3 The lower bound

In this section we derive a lower bound on the size of the kernel for the 3-3-HS problem on 3-uniform hypergraphs.

Using the techniques in [6], Cai proved that unless $P=NP$, 3-HS on 3-uniform hypergraphs of degree at most Δ , has no kernel of size at most $(1+1/(\sqrt{2\Delta+1}-1)-\epsilon)k$, for any $\epsilon > 0$ [4]. Assuming that $P \neq NP$, Cai's result implies a lower bound of $(1+1/(\sqrt{7}-1)-\epsilon)k < 1.608k$ on the kernel size of 3-3-HS on 3-uniform hypergraphs. Using the results in [6], and a more accurate analysis than that performed in [4], we shall improve on this lower bound next.

An *independent set* in a hypergraph \mathcal{H} is a set of vertices $I \subseteq V(\mathcal{H})$ such that no edge in \mathcal{H} is completely contained in I ; that is, for every edge $e \in E(\mathcal{H})$: $e \cap (V \setminus I) \neq \emptyset$. The INDEPENDENT SET problem on hypergraphs is: given a hypergraph \mathcal{H} and a nonnegative integer k , decide if there is an independent set in \mathcal{H} of size at least k . It can be readily seen that a set $I \subseteq V(\mathcal{H})$ is an independent set in \mathcal{H} if and only if $V(\mathcal{H}) \setminus I$ is a hitting set of \mathcal{H} . Therefore, the HITTING SET and the INDEPENDENT SET problem on hypergraphs are *dual* problems [6], in the same sense that VERTEX COVER and INDEPENDENT SET are dual problems on graphs.

The results of Caro and Tuza [5] imply that, for any 3-uniform hypergraph \mathcal{H} of degree at most 3, the independence number of \mathcal{H} (i.e., the size of a maximum independent set in \mathcal{H}), denoted $\alpha(\mathcal{H})$, satisfies the following inequality:

$$\alpha(\mathcal{H}) \geq \sum_{v \in V(\mathcal{H})} \prod_{i=1}^{\deg(v)} \frac{2i}{2i+1}. \quad (1)$$

Theorem 1. *The INDEPENDENT SET problem on 3-uniform hypergraphs of degree at most 3 has a kernel of size $35k/16$ that is computable in $O(k)$ time.*

Proof. Inequality (1) implies that the independence number of a 3-uniform hypergraph \mathcal{H} of degree at most 3 satisfies:

$$\alpha(\mathcal{H}) \geq (16/35)|V(\mathcal{H})|. \quad (2)$$

Consider the following kernelization algorithm for INDEPENDENT SET on 3-uniform hypergraphs of degree at most 3: Given an instance (\mathcal{H}, k) , if $|V(\mathcal{H})| \geq (35/16)k$ then accept; otherwise, output the original instance as the kernel.

Clearly, the above algorithm can be implemented to run in time linear in k . The correctness of the algorithm, as well as the upper bound on the kernel size, follow from Inequality (2). \square

Using the notion of duality introduced in [6], and the fact that the INDEPENDENT SET on hypergraphs and the HITTING SET are dual problems, Theorem 1 implies the following result:

Theorem 2. *Unless $P=NP$, the 3-3-HS problem on 3-uniform hypergraphs does not have a kernel of size at most $35k/19 > 1.8421k$.*

4 The kernel

In this section we present a kernelization algorithm for 3-3-HS. Note that we do not assume that the hypergraph is 3-uniform. As we showed in the previous section, unless $P=NP$, no kernel of size at most $(35/19)k > 1.842k$ exists for the 3-3-HS problem. A kernel of size at most $4k$ for the 3-3-HS problem is implied from the results in [15]. We shall improve on the $4k$ upper bound on the kernel size for 3-3-HS.

The following reduction operations are folklore (see [15], for example), and can be easily verified by the reader:

Reduction Rule 1: If there is a 1-edge $e = \{v\}$ then include v in the solution set S , set $\mathcal{H} := \mathcal{H} - e$ and $k := k - 1$, and decrement k by 1.

Reduction Rule 2: If edge e is dominated by edge e' then set $\mathcal{H} := \mathcal{H} - e'$.

Reduction Rule 3: If vertex u is dominated by vertex v then set $\mathcal{H} := \mathcal{H} - u$.

We assume that we have a subroutine **Reduce** (\mathcal{H}, k) that applies **Reduction Rules 1–3** to the instance (\mathcal{H}, k) . We say that the instance (\mathcal{H}, k) is *reduced* if none of **Reduction Rules 1–3** applies to (\mathcal{H}, k) . We shall assume in what follows that the instance (\mathcal{H}, k) is reduced.

Definition 1. An edge $e \in \mathcal{H}$ is *good* if it contains exactly one degree-3 and two degree-2 vertices; otherwise, edge e is *bad*. A vertex $v \in \mathcal{H}$ is *good* if every edge containing v is good; otherwise, vertex v is *bad*.

Before we present the technical results of this section, we briefly and intuitively describe the ideas behind these results.

We will show that if the number of bad vertices is “large”, say larger than a certain function $g(k)$, then the size of the instance has to be at most $4k - g(k)/6$, in order for a solution (i.e., a hitting set of size at most k) to exist (Lemma 6), thus improving on the $4k$ upper bound on the kernel size in this case.⁴ This allows us to upper bound the number of bad vertices in the instance. On the other hand, we show that if the number of degree-2 vertices in the solution is “large”, say at least $g(k)$, then the size of the instance must be upper bounded by $4k - g(k)$ (Lemma 7), again improving on the $4k$ upper bound in this case. We then proceed to show that if a degree-2 good vertex whose distance from *every* bad vertex is more than some positive integer h , is contained in *every* solution, then this vertex forces at least $2^{\lfloor h/2 \rfloor + 2} - 3$ degree-2 vertices to be in a solution (Lemma 9), thus upper bounding the size of the instance by $4k - 2^{\lfloor h/2 \rfloor + 2} + 3$ (Lemma 7). Therefore, if the size of the instance is larger than $4k - 2^{\lfloor h/2 \rfloor + 2} + 3$, then no degree-2 good vertex whose distance is more than h from every bad vertex can be contained in every solution, and hence, any such vertex can be discarded from the instance (Lemma 10). After discarding all such vertices, every good degree-2 vertex (resp. good degree-3 vertex) must be within distance $h + 1$ (resp. $h + 2$) from some bad vertex. Since the number of bad vertices has been upper bounded by $g(k)$, this allows us to derive an upper bound on the total number of vertices (both good and bad), and hence on the size of the instance. By choosing $g(k)$ and h appropriately, we can derive an upper bound of $4k - k^{0.2692}$ on the size of the kernel. We now proceed to the technical details.

The following lemma follows from Definition 1:

Lemma 1. *No two degree-3 good vertices are adjacent.*

Lemma 2. *Let u be a good degree-2 vertex, and let v and w be its degree-3 neighbors. If every minimum hitting set of \mathcal{H} contains u , then no minimum hitting set of \mathcal{H} contains v or w .*

Proof. If there is a minimum hitting set that contains both u and v (resp. u and w), then we can replace u by w (resp. v) to obtain a minimum hitting set that does not contain u , contradicting the hypothesis. \square

Lemma 3. *A degree-3 good vertex has exactly 6 degree-2 neighbors.*

Proof. Let v be a degree-3 good vertex, and let edges $\{v, x, y\}$, $\{v, u, w\}$, $\{v, p, q\}$ be the three edges containing v .

By Definition 1, vertices x, y, u, w, p, q are all of degree 2. Therefore, it suffices to show that all these vertices are distinct. Suppose not, then there exists a vertex among x, y, u, w, p, q that is dominated by v . This contradicts the fact that the instance (\mathcal{H}, k) is reduced. \square

⁴ It may sound counterintuitive to call such vertices “bad” since they allow us to upper bound the size of the instance. However, as will be shown later, it turns out that the existence of such vertices, and more specifically, the proximity of other vertices to the bad vertices, is what prohibits us from simplifying the instance further.

The following lemma is straightforward.

Lemma 4. *Let u be a good degree-3 vertex. If a hitting set S excludes u , then S must contain at least one degree-2 vertex from every edge containing u , and hence, at least three distinct degree-2 vertices that are neighbors of u must be in S .*

Lemma 5. *Let S be a hitting set of \mathcal{H} . Then there exists a hitting set S' of \mathcal{H} of size at most $|S|$ such that: (a) no two degree-2 good vertices in S' are neighbors, and (b) no good degree-2 vertex in S' is a neighbor of a degree-3 vertex in S' .*

Proof. Among all hitting sets of \mathcal{H} of size at most $|S|$, let S' be one with the minimum number of good degree-2 vertices. We claim that S' satisfies properties (a) and (b) above.

Suppose that S' does not satisfy property (a), and let u and v be two good degree-2 vertices in S' that are neighbors. Let $e = \{u, v, x\}$ be the edge containing u and v , and let $e' = \{v, w, y\}$ be the other edge containing v . Since u and v are good vertices, e and e' are good edges. From the definition of a good edge, it follows that x is a degree-3 vertex, and exactly one vertex in $\{w, y\}$, say y , must be of degree 3. By excluding v from S' and including y (if y is not already included), we obtain a hitting set of \mathcal{H} of size at most $|S'| \leq |S|$, in which the number of degree-2 good vertices is strictly less than that of S' , contradicting the minimality of S' . This proves part (a).

To prove that S' satisfies part (b), suppose not, and let v be a good degree-2 vertex in S' that is a neighbor of a degree-3 vertex x in S' . Let $e = \{u, v, x\}$ and $e' = \{v, w, y\}$ be the good edges containing v , and assume, without loss of generality, that y is of degree 3. Then by replacing v in S' by y , we obtain a hitting set of size at most $|S'| \leq |S|$ that contains fewer degree-2 good vertices than S' , contradicting the minimality of S' . This proves the lemma. \square

Let $g(k)$ be a function of k to be determined later.

Lemma 6. *If the number of bad vertices in \mathcal{H} is at least $g(k)$, then either $|V| \leq 4k - g(k)/6$, or \mathcal{H} does not have a hitting set of size at most k .*

Proof. Suppose that \mathcal{H} has a hitting set S of size at most k . The set of bad edges in \mathcal{H} can be partitioned into the following sets:

1. the set of 2-edges, denoted E_2 ;
2. the set of 3-edges whose vertices are all of degree 2, denoted E_3^2 ; and
3. the set of 3-edges that each contains at least two degree-3 vertices, denoted E_3 .

We define an *occurrence* of a vertex v to be an edge e that contains v . Clearly, the number of occurrences of a vertex is equal to its degree. We call the occurrences of vertices in S the *normal occurrences*, and those of vertices not in S the *extra occurrences*. We count the total number of extra occurrences next.

Let $E'_3 \subseteq E_3$ be the set of 3-edges that contain at least two vertices in S , and let $E''_3 = E_3 \setminus E'_3$.

Each edge in E_3^2 forces at least one degree-2 vertex to be in S , and at most 2 edges in E_3^2 can be covered by the same degree-2 vertex in S . On the other hand, an edge in E'_3 forces two vertices in S to cover the same edge. Since S has size at most k , and since the degree of \mathcal{H} is at most 3, it follows from the previous statements that the total number of edges that S can cover is at most $3k - |E_3^2|/2 - |E'_3|$. Since each of these edges contains at least one vertex from S , and since each edge in E_2 has size 2, the number of extra occurrences is at most $2(3k - |E_3^2|/2 - |E'_3|) - |E_2|$.

Since every vertex in \mathcal{H} has degree at least 2, every vertex not in S must contribute at least 2 extra occurrences. Moreover, since each edge in E''_3 contains at least one degree-3 vertex that is not in S , at least one vertex in each edge of E''_3 contributes 3 extra occurrences. Therefore, the number of vertices not in S , i.e. $|V(\mathcal{H}) \setminus S|$, is at most:

$$\begin{aligned} |V(\mathcal{H}) \setminus S| &\leq (2(3k - |E_3^2|/2 - |E'_3|) - |E_2| - |E''_3|)/2 \\ &= 3k - |E_3^2|/2 - |E'_3| - |E_2|/2 - |E''_3|/2 \\ &= 3k - (|E_3^2|/2 + |E'_3|/2 + |E''_3|/2 + |E_2|/2) - |E'_3|/2 \\ &\leq 3k - (|E_3^2|/2 + |E_3|/2 + |E_2|/2). \end{aligned} \tag{3}$$

The number of bad edges is $|E_3^2| + |E_3| + |E_2|$. Moreover, since each bad edge can induce at most 3 bad vertices, the number of bad vertices is at most 3 times the number of bad edges, and hence $|E_3^2| + |E_3| + |E_2| \geq g(k)/3$. Combining the last inequality with Inequality (3), we derive that $|V(\mathcal{H}) \setminus S| \leq 3k - g(k)/6$.

It follows that if \mathcal{H} has a hitting set of size at most k then $|V(\mathcal{H})| = |V(\mathcal{H}) \setminus S| + |S| \leq 4k - g(k)/6$. This completes the proof. \square

Lemma 7. *Suppose that \mathcal{H} has a hitting set S of size at most k . Let S_2 be the set of degree-2 vertices in S . If $|S_2| \geq g(k)$, then $|V| \leq 4k - g(k)$.*

Proof. Let S_3 be the set of degree-3 vertices in S . The number of edges that S can cover is at most $2|S_2| + 3|S_3|$. Since S is a hitting set, S covers all edges in \mathcal{H} . Since each edge must contain at least one vertex from S , the total number of extra occurrences (defined in the proof of Lemma 6) of all vertices is at most $4|S_2| + 6|S_3|$. Each vertex not in S contributes at least 2 extra occurrences. Therefore, the number of vertices in $V \setminus S$ is at most $2|S_2| + 3|S_3| = 3|S| - |S_2| \leq 3k - |S_2| \leq 3k - g(k)$. It follows that $|V| = |V \setminus S| + |S| \leq 4k - g(k)$. \square

Let h be a nonnegative integer, and let v be a degree-2 good vertex whose distance from every bad vertex is at least $h + 1$. Suppose that v is contained in every minimum hitting set of \mathcal{H} , and let S be a minimum hitting set of \mathcal{H} . By Lemma 5, we can assume that no two good degree-2 vertices in S are neighbors, and that no good degree-2 vertex in S is a neighbor of a degree-3 vertex in S . We define a layered graph T_v (we prove next that T_v is a tree rooted at v), with

respect to the minimum hitting set S . The graph T_v consists of exactly $h + 1$ layers L_0, \dots, L_h , defined as follows. Layer L_0 consists of the vertex v . For odd $i \in \{1, \dots, h\}$, layer L_i consists of the degree-3 (good) vertices that are neighbors of the vertices in L_{i-1} , and that do not appear in a previous layer L_j for $j < i$. For even $i \in \{1, \dots, h\}$, layer L_i consists of the good degree-2 neighbors of the vertices in layer L_{i-1} that are in S , and that do not appear in a previous layer L_j for $j < i$. There is an edge between two vertices in T_v if and only if they are neighbors in \mathcal{H} . The following lemma describes the structural properties of T_v .

Lemma 8. *The following are true:*

- (i) *Vertex v has exactly two good degree-3 neighbors that form layer L_1 .*
- (ii) *No two degree-2 vertices in T_v are adjacent, and no two degree-3 vertices in T_v are adjacent.*
- (iii) *For odd $i \in \{0, \dots, h\}$, every vertex in layer L_i is a good degree-3 vertex that is not in S , and that has exactly one neighbor in layer L_{i-1} . For even $i \in \{1, \dots, h\}$, every vertex in layer L_i is a good degree-2 vertex that is in S , and that has exactly one neighbor in layer L_{i-1} .*
- (iv) *For odd $i \in \{0, \dots, h-1\}$, every vertex in layer L_i has two neighbors in layer L_{i+1} . For even $i \in \{1, \dots, h-1\}$, every vertex in layer L_i has exactly one neighbor in layer L_{i+1} . Moreover, two distinct vertices in layer L_i have distinct neighbors in layer L_{i+1} .*
- (v) *T_v is a tree.*

By Lemma 8, T_v is a tree rooted at v . Therefore, we can now refer to the *parent* of a vertex $w \in T_v$, denoted $\pi(w)$, and the *children* of w , in the usual sense.

Lemma 9. *Let n_2 be the number of degree-2 vertices in T_v . Then $n_2 = 2^{\lfloor h/2 \rfloor + 2} - 3$.*

Proof. Note that the degree-2 vertices appear only in even layers of T_v . Layer L_0 contains exactly one degree-2 vertex, namely v . By Lemma 8, every vertex in an even layer L_i , $0 < i < h$, has exactly one child in layer L_{i+1} , and every vertex in an odd layer L_i , $0 < i < h$, has exactly two children in layer L_{i+1} . Therefore, we can write the following recurrence to describe the number of vertices in layer L_{2i} , denoted $|L_{2i}|$, for $i = 0, \dots, \lfloor h/2 \rfloor$:

$$|L_{2i}| = \begin{cases} 1 & \text{if } i = 0, \\ 4 & \text{if } i = 1, \\ 2|L_{2i-2}| & \text{for } i = \{2, \dots, \lfloor h/2 \rfloor\}. \end{cases}$$

Solving the above recurrence relation, we obtain $|L_{2i}| = 2^{i+1}$, for $i = \{1, \dots, \lfloor h/2 \rfloor\}$. The total number of degree-2 vertices in T_v , n_2 , is then:

$$n_2 = |L_0| + \sum_{i=1}^{\lfloor h/2 \rfloor} |L_{2i}| = 1 + \sum_{i=1}^{\lfloor h/2 \rfloor} 2^{i+1} = 1 + 4(2^{\lfloor h/2 \rfloor} - 1) = 2^{\lfloor h/2 \rfloor + 2} - 3.$$

□

Lemma 10. *Let h be a nonnegative integer, and let v be a good degree-2 vertex whose distance from every bad vertex is at least $h + 1$. If $|V| > 4k - 2^{\lfloor h/2 \rfloor + 2} + 3$, then there is minimum hitting set of \mathcal{H} that does not contain v .*

Proof. Proceed by contradiction. Suppose that every minimum hitting set contains v . Consider a minimum hitting set S of \mathcal{H} , and assume, without loss of generality, that S satisfies the properties in Lemma 5. We construct the tree T_v as described above. By Lemma 9, the number of degree-2 vertices that are in T_v , and hence in S , is $2^{\lfloor h/2 \rfloor + 2} - 3$. By Lemma 7, $|V| \leq 4k - 2^{\lfloor h/2 \rfloor + 2} + 3$, contradicting the hypothesis in the statement of the theorem. \square

Lemma 11. *Let B be the set of bad vertices in \mathcal{H} . If every degree-2 good vertex in \mathcal{H} is of distance at most $h + 1$ from some bad vertex in B , then $|V(\mathcal{H})| \leq 4|B|2.562^{h+1} + |B|$.*

Consider the following algorithm:

Algorithm: Kernel-3-3-HS (\mathcal{H}, k)

if $|V(\mathcal{H})| > 9k$ **then reject**;
else repeat the following:

0. call **Reduce**(\mathcal{H}, k); **if** $|V(\mathcal{H})| > 4k$ **then reject**;
1. **if** $|V(\mathcal{H})| \leq 4k - g(k)$ **then return** the resulting instance;
2. let B be the set of bad vertices in \mathcal{H} ; **if** $|B| > 6g(k)$ **then reject**;
3. grow a Breadth-First Search (BFS) forest \mathcal{F} rooted at the vertices in B and stop at depth $h(k)$;
4. **if** all the degree-2 vertices in \mathcal{H} are also in \mathcal{F} **then return** the resulting instance;
5. let v be a degree-2 vertex in $V(\mathcal{H}) - V(\mathcal{F})$; remove v from \mathcal{H} ;

To optimize the upper bound on the size of the kernel, we choose $g(k) = k^{0.2692}$ and $h(k) = \log_{3.6235} k$ in the algorithm **Kernel-3-3-HS**. Let (\mathcal{H}', k') be the instance returned by the algorithm **Kernel-3-3-HS**.

Theorem 3. *Given an instance (\mathcal{H}, k) of 3-3-HS, in time $O(k^{1.2692})$ the algorithm **Kernel-3-3-HS** returns an equivalent instance (\mathcal{H}', k') such that $|V(\mathcal{H}')| \leq 4k' - k'^{0.2692}$.*

Proof. Observe that since the size of each edge in \mathcal{H} is at most 3, and since every vertex in \mathcal{H} has degree at most 3, the total number of edges and vertices in \mathcal{H} must be at most $9k$ if a solution of size k exists. Otherwise, we can reject the original instance directly. With this observation in mind, it is not difficult to see that the subroutine **Reduce**(\mathcal{H}, k) can be implemented to run in $O(k)$ time using the proper data structures. If after the application of **Reduce** more than $4k$ vertices remain in \mathcal{H} , then the instance can be rejected by Lemma 6. Clearly, steps 1–5 of the algorithm can be implemented to run in $O(k)$ time.

Therefore, each execution of steps 0–5 of the algorithm takes $O(k)$ time. After the first application of **Reduce**, at most $4k$ vertices remain in \mathcal{H} or the instance is rejected. Since in each execution the algorithm either returns a kernel (step 1 or step 4), rejects the instance (step 2), or removes a vertex from \mathcal{H} (step 5), and since the algorithm stops once the number of vertices in \mathcal{H} is at most $4k - k^{0.2692}$, the number of executions of steps 0–5 is at most $k^{0.2692}$. It follows that the algorithm runs in $O(k^{1.2692})$ time.

To prove the correctness of the algorithm, note that since **Reduction Rules 1–3** are sound, the subroutine **Reduce**(\mathcal{H}, k) is correct, and step 1 of the algorithm is valid. If in step 2 $|B| > 6k^{0.2692}$, then since $|V| > 4k - k^{0.2692}$ (from step 1), it follows from Lemma 6 that \mathcal{H} does not have a hitting set of size at most k , and the algorithm can reject the instance. Therefore, step 2 is correct. If the algorithm removes a vertex v in step 5, then since $v \in V(\mathcal{H}) - V(\mathcal{F})$, the distance between v and any bad vertex in B is more than $\log_{3.6235} k$. Since $|V| > 4k - k^{0.2692}$, by Lemma 10, there is a solution that excludes v , and hence v can be safely removed from \mathcal{H} . It follows that step 5 is correct, and so is the algorithm **Kernel-3-3-HS**. Therefore, the instance (\mathcal{H}', k') returned by the algorithm is equivalent to the instance (\mathcal{H}, k) .

To prove that the algorithm returns an instance of size at most $4k' - k'^{0.2692}$, note that the algorithm returns an equivalent instance only in steps 2 and 4. Clearly, if the algorithm returns an instance in step 2 then the size of the instance is at most $4k' - k'^{0.2692}$. If the algorithm returns an instance in step 4, then the number of vertices in \mathcal{H} is bounded by $4|B|2.562^{\log_{3.6235} k'+1} + |B| < 9.076|B|k'^{0.73075} + |B| < 4k' - k'^{0.2692}$.

Therefore we conclude that the size of the instance is at most $4k' - k'^{0.2692}$. This completes the proof. \square

5 Generalization to bounded degree Δ

The kernelization results in the previous section can be generalized to hypergraphs of degree at most Δ , for $\Delta > 3$. We modify the definition of bad and good edges and vertices as follows. Let (\mathcal{H}, k) be an instance of 3-HS, where H has degree at most Δ . An edge e is *good* if e is a 3-edge in which exactly two vertices are of degree 2 and the third vertex is of degree more than 2; otherwise, e is *bad*. A vertex v is *good* if every edge containing v is good; otherwise, v is *bad*. With the modified definition of good and bad edges and vertices, and using a parallel approach to the one used in the previous section, we can show the following:

Theorem 4. *The 3-HS problem on hypergraphs of degree at most $\Delta > 3$ has a kernel of size $4k - O(k^{\frac{1}{2+4\log(\Delta-1)}})$ that is computable in time $O(k^{1+\frac{1}{2+4\log(\Delta-1)}})$.*

6 Concluding remarks

In this paper we gave upper and lower bounds on the kernel size for 3-3-HS. Although our improvement on the upper bound of the kernel size for 3-3-HS is

small, the techniques involved are highly nontrivial. This hints at the level of difficulty of the problem, and may suggest that a linear improvement on the kernel size for 3-3-HS may not be easy. We leave this as an open problem.

References

1. F. Abu-Khzam. Kernelization algorithms for d -hitting set problems. In *WADS*, pages 434–445, 2007.
2. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial optimization problems and their approximability properties*. Springer, 1999.
3. R. Bar-Yehuda and S. Even. A local-ratio theorem for approximating the Weighted Vertex Cover problem. *Annals of Discrete Mathematics*, 25:27–46, 1985.
4. X. Cai. Linear kernelizations for restricted 3-hitting set problems. *Inf. Process. Lett.*, 109(13):730–738, 2009.
5. Y. Caro and Z. Tuza. Improved lower bounds on k -independence. *Journal of Graph Theory*, 15:99–107, 1991.
6. J. Chen, H. Fernau, I. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. *SIAM J. Comput.*, 37(4):1077–1106, 2007.
7. J. Chen, I. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *Journal of Algorithms*, 41:280–301, 2001.
8. J. Chen, I. Kanj, and G. Xia. Labeled search trees and amortized analysis: improved upper bounds for NP-hard problems. *Algorithmica*, 43(4):245–273, 2005.
9. J. Chen, I. Kanj, and G. Xia. Improved parameterized upper bounds for vertex cover. volume 411, pages 3736–3756, 2010.
10. R. Downey and M. Fellows. *Parameterized Complexity*. Springer, New York, 1999.
11. H. Fernau. A top-down approach to search-trees: Improved algorithmics for 3-hitting set. *Electronic Colloquium on Computational Complexity (ECCC)*, (073), 2004.
12. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, 1979.
13. R. Niedermeier and P. Rossmanith. An efficient fixed-parameter algorithm for 3-hitting set. *J. Discrete Algorithms*, 1(1):89–102, 2003.
14. I. Razgon. Faster computation of maximum independent set and parameterized vertex cover for graphs with maximum degree 3. *J. Discrete Algorithms*, 7(2):191–212, 2009.
15. M. Wahlström. Algorithms, measures, and upper bounds for satisfiability and related problems, 2007. Ph.D. Thesis (Linköping Studies in Science and Technology, PhD Dissertation no 1079). Available at <http://liu.diva-portal.org/smash/record.jsf?pid=diva2:23420>.
16. D. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.